

The Anderson-Moore Algorithm: Numeric Mathematica Implementation

Gary Anderson

August 31, 1998

Abstract

(?; ?) describe a powerful method for solving linear saddle point models. The algorithm has proved useful in a wide array of applications including analyzing linear perfect foresight models, providing initial solutions and asymptotic constraints for nonlinear models. The algorithm solves linear problems with dozens of lags and leads and hundreds of equations in seconds. The technique works well for both symbolic algebra and numerical computation.

Although widely used at the Federal Reserve, few outside the central bank know about or have used the algorithm. This paper attempts to present the current algorithm in a more accessible format in the hope that economists outside the Federal Reserve may also find it useful. In addition, over the years there have been undocumented changes in approach that have improved the efficiency and reliability of algorithm. This paper describes the present state of development of this set of tools.

1 Problem Statement

Anderson and Moore (?) outlines a procedure that computes solutions for structural models of the form

$$\sum_{i=-\tau}^{\theta} H_i x_{t+i} = 0 , \quad t \geq 0 \quad (1)$$

with initial conditions, if any, given by constraints of the form

$$x_t = x_t^{data} , \quad t = -\tau, \dots, -1 \quad (2)$$

where both τ and θ are non-negative, and x_t is an L dimensional vector with

$$\lim_{t \rightarrow \infty} x_t = 0 \quad (3)$$

The algorithm determines whether the model 1 has a unique solution, an infinity of solutions or no solutions at all.

The specification 1 is not restrictive. One can handle inhomogeneous version of equation 1 by recasting the problem in terms of deviations from a steady state value or by adding a new variable for each non-zero right hand side with an equation guaranteeing the value always equals the inhomogeneous value ($x_t^{con} = x_{t-1}^{con}$ and $x_{t-1}^{con} = x^{RHS}$).

Saddle point problems combine initial conditions and asymptotic convergence to identify their solutions. The uniqueness of solutions to system 1 requires that the transition matrix characterizing the linear system have an appropriate number of explosive and stable eigenvalues(?), and that the asymptotic linear constraints are linearly independent of explicit and implicit initial conditions(?).

The solution methodology entails

1. using equation 1 to compute a state space transition matrix.
2. Computing the eigenvalues and the invariant space associated with large eigenvalues
3. Combining the constraints provided by:
 - (a) the initial conditions,
 - (b) auxiliary initial conditions identified in the computation of the transition matrix and
 - (c) the invariant space vectors

Figure 1 presents a flow chart summarizing the algorithm. For a description of a parallel implementation see (?) For a description of a continuous application see (?).

2 Algorithm

2.1 Unconstrained Auto-regression

The algorithim in this section performs elementary row operations on the original model equations to produce a normal form that facilitates construction of a state space transition matrix. If the leading square block of the linear system(H_θ) were non-singular, one could compute the transition matrix from

$$A = \begin{bmatrix} 0 & I & & \\ 0 & & I & \\ 0 & & & \ddots \\ 0 & & & I \\ H_\theta^{-1} [H_{-\tau\alpha} \dots H_{\theta-1}] & & & \end{bmatrix}$$

Since H_θ is typically singular, the algorithm identifies linear combinations of the first $L(\tau + \theta + 1)$ equations that have leading block non-singular. Inverting this non-singular right hand block and multiplying the lagged matrices provides the autoregressive representation. The remaining equations, the *auxiliary initial*

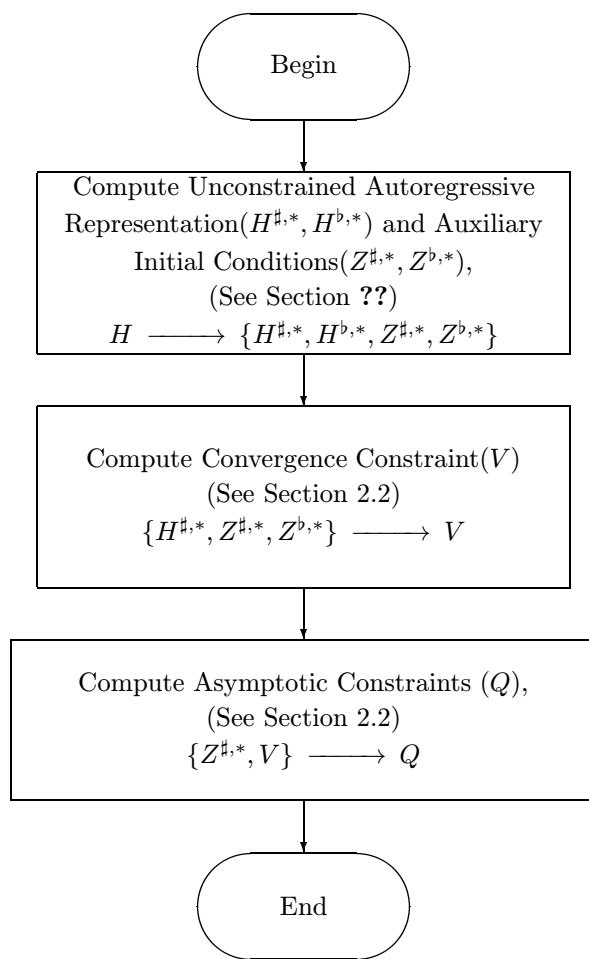


Figure 1: Algorithm Overview

conditions, provide important information for determining a unique trajectory for the model. Section 2.3 shows how the auxiliary initial conditions provide a mechanism for reducing the size of the eigen system calculation.

Algorithm 1

```

1 Given  $H$ , compute the unconstrained auto-regression.
2 funct  $\mathcal{F}_1(H)$   $\equiv$ 
3    $k := 0$ 
4    $\mathcal{Z}^0 := \emptyset$ 
5    $\mathcal{H}^0 := H$ 
6   while  $\mathcal{H}_\theta^k$  is singular  $\cap \text{rows}(\mathcal{Z}^k) < L(\tau + \theta)$ 
7     do
8        $U^k = \begin{bmatrix} U_Z^k \\ U_N^k \end{bmatrix} := \text{rowAnnihilator}(\mathcal{H}_\theta^k)$ 
9        $\mathcal{H}^{k+1} := \begin{bmatrix} 0 & U_Z^k \mathcal{H}_\tau^k & \dots & U_Z^k \mathcal{H}_{\theta-1}^k \\ U_N^k \mathcal{H}_\tau^k & \dots & U_N^k \mathcal{H}_\theta^k \end{bmatrix}$ 
10       $\mathcal{Z}^{k+1} := \begin{bmatrix} \mathcal{Q}^k \\ U_Z^k \mathcal{H}_\tau^k & \dots & U_Z^k \mathcal{H}_{\theta-1}^k \end{bmatrix}$ 
11       $k := k + 1$ 
12    od
13  return{ $[\mathcal{H}_{-\tau}^k \dots \mathcal{H}_\theta^k]$ , ( $\Gamma$  or  $\emptyset$ ),  $\mathcal{Z}^k$ }
14 .

```

Theorem 1 Let

$$\mathcal{H} = \left[\begin{array}{ccccccccc} H_{-\tau} & & \dots & & H_\theta & & & & \\ & H_{-\tau} & & \dots & & H_\theta & & & \\ & & \ddots & & & & & & \\ & & & H_{-\tau} & & \dots & & H_\theta & \\ & & & & H_{-\tau} & & \dots & & H_\theta \\ & & & & & H_\theta & & & \\ & & & & & & \ddots & & \\ & & & & & & & \ddots & \\ & & & & & & & & H_\theta \end{array} \right]_{\tau+\theta+1}$$

There are two cases:

- When \mathcal{H} is full rank the algorithm terminates with $Z^{\sharp*}(Z^{\flat*})$ and non-singular $H_\theta^{\sharp*}(H_\tau^{\flat*})$
- When \mathcal{H} is not full rank the algorithm terminates when some row of $[\mathcal{H}_{-\tau} \dots \mathcal{H}_\theta]$ is zero.

2.1.1 numericRightMostAllZeroQ

This routine checks the \dim right most elements of a vector numbers(x) to see if they are small. The routine compares $\|x_{n-\dim+1} \dots x_i\|_2$ to $\$zeroTol$ and returns *False* if it is larger and *True* otherwise.

```

⟨numericRightMostAllZeroQMathematica 1⟩ ≡

$zeroTol=10^(-10)

numericRightMostAllZeroQ[dim_,x_]:= 
  With[{lilvec=Take[x,-dim]},
  If[Apply[And,(Map[NumberQ,lilvec])],
  (Inner[Times,lilvec,lilvec,Plus]/(dim*dim))<=$zeroTol,
  If[Apply[And, (Map[Simplify[#] === 0&,lilvec])],True,False]]]

```

◇

Macro referenced in scrap 17.

2.1.2 numericShiftRightAndRecord

This routine implements lines 9-10 of the algorithm. It takes a set of auxiliary initial conditions and an input matrix(H) and

- identifies rows of H with zeroes in the rightmost block
- shifts these rows right one block
- adds the shifted row to the list of auxiliary conditions

The routine throws an exception if it identifies a matrix row that is all zeroes.

⟨numericShiftRightAndRecordMathematica 2⟩ ≡

```

numericShiftRightAndRecord[{auxiliaryConditionsSoFar_,hMatPreShifts_}]:= 
  If[Apply[Or,Map[Function[x,Apply[And ,Map[#=0&,x]]],hMatPreShifts]],
  Throw[{auxiliaryConditionsSoFar,hMatPreShifts},$zeroRow],
  With[{dim=Length[hMatPreShifts],ldim=Length[hMatPreShifts[[1]]]},
  FoldList[If[numericRightMostAllZeroQ[dim,#2],
  {Append[#1[[1]],Drop[#2,-dim]],
  Append[#1[[2]],RotateRight[#2,dim]]},
  {#[[1]],Append[#1[[2]],#2]}]&,
  {auxiliaryConditionsSoFar,{}} ,hMatPreShifts][[-1]]]]

```

◇

Macro referenced in scrap 17.

2.1.3 numericComputeAnnihilator

This routine uses QR Decomposition with column pivoting to construct a matrix that zeroes the rows of the input matrix. Mathematica uses QR Decomposition to determine the $\text{rank}(r)$ of A and returns an orthogonal matrix $Q \ni R = Q^T A$ has r rows. The routine extends the matrix to a basis for the n dimensional space by computing the orthogonal complement of Q , (\bar{Q}).

$$\begin{bmatrix} Q^T \\ \bar{Q}^T \end{bmatrix} A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Since Mathematica actually computes the transpose of the matrix Q , the routine does not have to transpose the matrix.

$\langle \text{numericComputeAnnihilatorMathematica} \ 3 \rangle \equiv$

```
numericComputeAnnihilator[amat_]:=With[{adim=Length[amat],
  fa=Flatten[amat]},If[Apply[And,(Map[NumberQ,fa])]&&(Max[Abs[fa]]<= $zeroTol),
  IdentityMatrix[Length[amat]],With[{qr=
  Select[QRDecomposition[amat,Pivoting->True][[1]],
  (Not[Apply[And,(Map[NumberQ,#])]]||(Max[Abs[#]]>0))&},
  If[Length[qr]==adim,IdentityMatrix[adim],
  Join[qr,Drop[
  Apply[Join,numericExtendToBasis[qr]],
  Length[qr]]]]]]];
```

◇

Macro referenced in scrap 17.

2.1.4 numericAnnihilateRows

This routine zeroes out as many rows as possible in the rightmost block rows of the matrix $hmat$. The $hmat$ will remain unchanged if the rightmost block is non-singular.

$\langle \text{numericAnnihilateRowsMathematica} \ 4 \rangle \equiv$

```
numericAnnihilateRows[hmat_]:=With[{dims=Dimensions[hmat]},
  With[{zapper=
  numericComputeAnnihilator[
  SubMatrix[hmat,{1,dims[[2]]-dims[[1]]+1},dims[[1]]{1,1}]]},
  If[zapper=={},hmat,zapper . hmat]]]
```

◇

Macro referenced in scrap 17.

2.1.5 numericAR

This routine implements Algorithm 1.

$\langle \text{numericARMathematica} \ 5 \rangle \equiv$

```

numericAR[hmat_]:= 
Catch[
FixedPoint[numericShiftRightAndRecord[
  {#[[1]], numericAnnihilateRows[#[[2]]]}]&,
{{}, hmat}, Length[hmat[[1]]],
SameTest->(Length[#1[[1]]] ==Length[#2[[1]]]&),
$zeroRow,$rankDeficiency[#1]&
]
]

```

Macro referenced in scrap 17.

2.1.6 numericBiDirectionalAR

This routine applies Algorithm 1 in the forward and the reverse directions to provide inputs for the state space reduction routines. The auxiliary initial conditions span the invariant space associated with zero eigenvalues. Obtaining these vectors makes it possible to compute the minimal dimension transition matrix.

$\langle \text{numericBiDirectionalARMathematica} \ 6 \rangle \equiv$

```

numericBiDirectionalAR[hmat_]:= 
{numericAR[hmat], Map[(Map[Reverse, #])&, numericAR[Map[Reverse, hmat]]]};

```

Macro referenced in scrap 17.

$\langle \text{numericAIMVersionMathematica} \ 7 \rangle \equiv$

```

numericAIMVersion[]:="$Revision: 1.3 $"

```

Macro referenced in scrap 17.

2.2 Invariant Space Calculations

Theorem 2 Let $\{x_t^{conv}\}$, $t = -\tau, \dots, \infty$ be a non explosive solution satisfying equation 1. Let A be the state space transition matrix for equation 1 and V be a set of invariant space vectors spanning the invariant space associated with roots of A of magnitude bigger than 1. Then for $t = 0, \dots, \infty$

$$V \begin{bmatrix} x_{t-\tau}^{conv} \\ \vdots \\ x_{t+\theta-1}^{conv} \end{bmatrix} = 0$$

Corrolary 1 Let $\{x_t\}$, $t = -\tau, \dots, \infty$ be a solution satisfying equation 1. If A has no roots with magnitude 1 then the path converges to the

unique steady state if and only if

$$V \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0$$

for some t .

Corrolary 2 If A has roots with magnitude 1 then a path converges to a limit cycle(or fixed point) if and only if

$$V \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0$$

for some t .

Algorithm 2

```

1 Given  $\Gamma^{\sharp,*}, Z^{\sharp,*}, Z^{\flat,*}$ ,
2 compute vectors spanning the left invariant
3 space associated with large eigenvalues
4 funct  $\mathcal{F}_2(\Gamma^{\sharp,*}, Z^{\sharp,*}, Z^{\flat,*})$ 
5    $A := \begin{bmatrix} 0 & I \\ \Gamma^{\sharp} & \end{bmatrix}$ 
6    $\{\bar{A}, \Pi, J_0\} = \text{stateSpaceReducer}(A, Z^{\sharp,*}, Z^{\flat,*})$ 
7    $\{\bar{V}, M\} := \text{leftInvariantSpaceVectors}(\bar{A})$ 
8    $V = \text{stateSpaceExpander}(\bar{V}, M, \Pi, J_0)$ 
9 .

```

2.2.1 computeAsymptoticConstraint

$\langle \text{numericAsymptoticConstraintMathematica} \rangle \equiv$

```

numericAsymptoticConstraint[zForward_, zBackward_, gammaForward_] :=
With[{squeezeResult = numericSqueeze[zForward, zBackward, gammaForward]},
With[{levs = Eigensystem[Transpose[squeezeResult[[3]]]]},
With[{lrg = Length[Select[levs[[1]], Abs[#] > 1 &]]},
With[{lrgvcs = Take[levs[[2]], lrg]},
Join[zForward, MapIndexed[(numericParticularLam[levs,
#2[[1]], squeezeResult[[2]]][[1]]) &,
Range[lrg]]]]]
◊

```

Macro referenced in scrap 17.

Theorem 3 Let

$$Q = \begin{bmatrix} Z^{\sharp} \\ V \end{bmatrix} = [Q_L \quad Q_R]$$

The existence of convergent solutions depends on the magnitude of the rank of the augmented matrix

$$r_1 = \text{rank} \left(\begin{bmatrix} I & 0 & x_{\text{data}} \\ Q_L & Q_R & 0 \end{bmatrix} \right)$$

and

$$r_2 = \text{rank} \left(\begin{bmatrix} I & 0 \\ Q_L & Q_R \end{bmatrix} \right)$$

and $L(\tau + \theta)$, the number of unknowns.

1. If $r_1 > r_2$ there is no nontrivial convergent solution
2. If $r_1 = r_2 = L(\tau + \theta)$ there is a unique convergent solution
3. If $r_1 = r_2 < L(\tau + \theta)$ the system has an infinity of convergent solutions

Corrolary 3 When Q has $L\theta$ rows, Q_R is square. If Q_R is non-singular, the system has a unique solution and

$$\begin{bmatrix} B \\ B_2 \\ \vdots \\ B_\theta \end{bmatrix} = Q_R^{-1} Q_L$$

If Q_R is singular, the system has an infinity of solutions.

Corrolary 4 When Q has fewer than $L\theta$ rows, The system has an infinity of solutions.

Corrolary 5 When Q has more than $L\theta$ rows, The system has a unique nontrivial solution only for specific values of x_{data}

Algorithm 3

```

1 Given  $V, Z^{\sharp,*}$ ,
2 funct  $\mathcal{F}_3(V, Z^{\sharp,*})$ 
3    $Q := \begin{bmatrix} Z^{\sharp,*} \\ V \end{bmatrix}$ 
4   cnt = noRows(Q)
5   return  $\begin{cases} \{Q, \infty\} & \text{cnt} < L\theta \\ \{Q, 0\} & \text{cnt} > L\theta \\ \{Q, \infty\} & (Q_R \text{ singular}) \\ \{-Q_R^{-1}Q, 1\} & \text{otherwise} \end{cases}$ 
6 .

```

2.3 State Space Reduction

Theorem 4 The Z_*^\sharp, Z_*^\flat span the invariant space associated with zero eigenvalue.

$$\begin{bmatrix} Z_*^\sharp \\ Z_*^\flat \end{bmatrix} A^{L(\tau+\theta)} = 0$$

Theorem 5 Suppose

$$Y\bar{A} = MY$$

so that Y spans the invariant space associated with the eigenvalues of M , one can compute X with

$$\begin{bmatrix} X & Y \end{bmatrix}$$

spans the dominant invariant space of A . From

$$\text{vec}(X) = ((I \otimes M) - (J_0^T \otimes I))^{-1} (\Pi^T \otimes I) \text{vec}(Y)$$

Algorithm 4

- 1 Given $h, H,$
- 2 asymptotic stability constraints
- 3 funct $\mathcal{F}_4(V, Z^{\sharp,*})$
- 4 .

$$\begin{aligned} Q_u^T RP &= [Z^{\sharp,*} Z^{\flat,*}]^T \\ Q_l^T RP &= Q_u^T Q_u - I \\ J_0 &= Q_u A Q_u^T \\ a &= Q_l A Q_l^T \\ \Pi &= Q_l A Q_u^T \end{aligned}$$

$\langle \text{numericTransitionMatrix} \rangle \equiv$

```
numericTransitionMatrix[transF_] :=
With[{nr=Length[transF]},
With[{nc=Length[transF[[1]]]-nr},
If[nc == nr,
-Inverse[SubMatrix[transF,{1,nc+1},{nr,nr}]].
SubMatrix[transF,{1,1},{nr,nc}],
BlockMatrix[{{ZeroMatrix[nc-nr,nr],IdentityMatrix[nc-nr]}, {-Inverse[SubMatrix[transF,{1,nc+1},{nr,nr}]]}.
SubMatrix[transF,{1,1},{nr,nc}]}],
]]
]

```

Macro referenced in scrap 17.

2.3.1 numericEliminateInessentialLags

$\langle \text{numericEliminateInessentialLags} \rangle \equiv$

```
numericEliminateInessentialLags[AMatrixVariableListPair_]:=  
  Block[{firstzerocolumn,matrixsize},  
    matrixsize=Length[AMatrixVariableListPair[[1]]];  
    firstzerocolumn=FindFirstZeroColumn[  
      AMatrixVariableListPair[[1]]];  
    Return[  
      If[IntegerQ[firstzerocolumn],  
        numericEliminateInessentialLags[  
          List[  
            ((AMatrixVariableListPair[[1]])[  
              Drop[Range[1,matrixsize],{firstzerocolumn}],  
              Drop[Range[1,matrixsize],{firstzerocolumn}]]],  
            Drop[AMatrixVariableListPair[[2]],{firstzerocolumn}]]],  
        AMatrixVariableListPair]]]  
  FindFirstZeroColumn[AMatrix_]:=Block[{columnnsumofabs},  
    columnnsumofabs=Map[Apply[And,#]&,  
      Map[Map[# == 0&,#]&, Transpose[AMatrix]]];  
    Return[Apply[Min,Position[columnnsumofabs,True]]]]
```

◇

Macro referenced in scrap 17.

2.3.2 numericSqueeze

$\langle \text{numericSqueezeMathematica 11} \rangle \equiv$

```
numericSqueeze[auxFAuxB_, transMat_] :=
Module[{evs, evcs, lamVal},
With[{extend=numericExtendToBasis[auxFAuxB]},(*{{p1,p2},mat1,mat2}*)
{Apply[Join,extend],extend[[1]] . transMat . Transpose[extend[[1]]],
extend[[2]] . transMat . Transpose[extend[[1]]],
extend[[2]] . transMat . Transpose[extend[[2]]]}]

(*
numericSqueeze[auxFAuxB_, transMat_] :=
Module[{evs, evcs, lamVal},
With[{extend=numericExtendToBasis[auxFAuxB]},(*{{p1,p2},mat1,mat2}*)
With[{r12= extend[[2]] . Transpose[extend[[1,2]]],
zeroDim=Length[extend[[1,1]]],
nonZeroDim=Length[extend[[1,2]]],
pmat=Join[extend[[1,1]],extend[[1,2]]]},
With[{ptrans=pmat . transMat. Transpose[pmat]},
{SubMatrix[ptrans,{1,1},zeroDim*{1,1}],
SubMatrix[ptrans,{(zeroDim+1),1},{nonZeroDim,zeroDim}],
SubMatrix[ptrans,(zeroDim+1){1,1},nonZeroDim*{1,1}] -
SubMatrix[ptrans,{(zeroDim+1),1},{nonZeroDim,zeroDim}] . r12 }]]]
*)
◊
```

Macro referenced in scrap 17.

2.3.3 numericMapper

$\langle \text{numericMapperMathematica 12} \rangle \equiv$

```
numericMapper[bigPi_,bigJ0_]:= 
Function[evMat,
  (Inverse[kron[IdentityMatrix[Length[bigJ0]],evMat] -
kron[Transpose[bigJ0],IdentityMatrix[Length[evMat]]]] .
kron[Transpose[bigPi],IdentityMatrix[Length[evMat]]]) ]
◊
```

Macro referenced in scrap 17.

2.3.4 numericEvExtend

$\langle \text{numericEvExtendMathematica } 13 \rangle \equiv$

```
(*  
numericEvExtend[evMat_,yMat_,mapper_,transf_]:=  
With[{xmat=mapper[evMat] . Transpose[{Flatten[Transpose[yMat]]}]},  
      BlockMatrix[{{Partition[Flatten[Transpose[xmat]],  
                  Length[xmat]/Length[evMat]],yMat}}] . transf]  
*)  
numericEvExtend[evMat_,yMat_,mapper_,transf_]:=  
With[{xmat=mapper[evMat] . Transpose[{Flatten[Transpose[yMat]]}]},  
      BlockMatrix[{{Transpose[Partition[Flatten[xmat],  
                  Length[yMat](*Length[xmat]/Length[evMat]*)],yMat}}] .  
      transf]
```

\diamond

Macro referenced in scrap 17.

2.3.5 numericParticularLam

$\langle \text{numericParticularLamMathematica } 14 \rangle \equiv$

```
numericParticularLam[es_List,n_Integer,eMap_,transf_]:=  
{Flatten[Append[eMap[{{es[[1,n]]}}] . Transpose[{es[[2,n]]}],es[[2,n]]]} . transf  
(*  
numericParticularLam[es_List,n_Integer,eMap_]:=Apply[eMap,Prepend[es[[2,n]],es[[1,n]]]]  
*)  
◊
```

Macro referenced in scrap 17.

2.3.6 numericExtendToBasis

$\langle \text{numericExtendToBasisMathematica } 15 \rangle \equiv$

```
numericExtendToBasis[matRows_List]:=  
With[{firstQR=QRDecomposition[Transpose[matRows],Pivoting->True]},  
With[{bigZ=Select[  
firstQR[[1]],  
      (Not[Apply[And,(Map[NumberQ,#])]]|| (Max[Abs[#]]>0))&]},  
With[{secondQR=QRDecomposition[Transpose[bigZ] . bigZ -  
      IdentityMatrix[Length[bigZ[[1]]]],Pivoting->True]},  
{bigZ,  
Select[  
secondQR[[1]],  
      (Not[Apply[And,(Map[NumberQ,#])]]|| (Max[Abs[#]]>0))&}]]]  
◊
```

Macro referenced in scrap 17.

AN EXAMPLE:

Collecting the auxiliary constraints generated by the auto regression phase of the algorithm for the example model one has:

$$\begin{bmatrix} Z_*^\sharp \\ Z_*^\flat \end{bmatrix} = \left[\begin{array}{ccccccccc} 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 \\ 0 & 0 & -\theta & 0 & 0 & -1 & 0 & 1 & 0 & -\gamma \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & \alpha & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

One can extend the basis to get a non singular matrix.¹

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix} = \left[\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{1}{\theta} & 0 & \frac{\gamma}{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -\frac{2\alpha}{\theta} & 2 & \frac{2\alpha\gamma-3\theta}{\theta} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$A = \begin{bmatrix} \rho & 2\gamma & -\gamma \\ 4\alpha & 3 & -2 \\ 2\alpha & 2 & -1 \end{bmatrix}$$

For this model, we expect one large root. Consequently, $M = [\lambda_L]$, a 1×1 matrix.

$$\Pi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -2\gamma \\ 0 & 0 & 0 & 0 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

$$J_0 = \left[\begin{array}{ccccccccc} 0 & 0 & 0 & 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 + \frac{4\alpha\gamma}{\theta} - \frac{2(2\alpha\gamma-3\theta)}{\theta} & \\ 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 0 \end{array} \right]$$

¹The top rows are just the row-echelon form of the $Z^{\sharp,*}, Z^{\flat,*}$ vectors.

$$((I \otimes M) - (J_0^T \otimes I))^{-1}(\Pi^T \otimes I) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-2\gamma}{\lambda_L} & \frac{-4}{\lambda_L} & \frac{-2}{\lambda_L} \end{bmatrix}$$

2.3.7 kron

$\langle \text{kronMathematica } 16 \rangle \equiv$

```
kron[a_,b_]:=BlockMatrix[Outer[Times,a,b]]
```

◇

Macro referenced in scrap 17.

2.3.8 numericLinearAim

$\langle \text{numericLinearAim } 17 \rangle \equiv$

```
BeginPackage["numericLinearAim`", {"LinearAlgebra`MatrixManipulation`"}]
<numericRightMostAllZeroQMathematica 1>
<numericShiftRightAndRecordMathematica 2>
<numericComputeAnnihilatorMathematica 3>
<kronMathematica 16>
<numericAnnihilateRowsMathematica 4>
<numericARMathematica 5>
<numericBiDirectionalARMathematica 6>
<numericTransitionMatrix 9>
<numericEliminateInessentialLagsMathematica 10>
<numericSqueezeMathematica 11>
<numericMapperMathematica 12>
<numericEvExtendMathematica 13>
<numericParticularLamMathematica 14>
<numericExtendToBasisMathematica 15>
<numericAsymptoticConstraintMathematica 8>
<numericAIMVersionMathematica 7>
EndPackage[]
◇
```

Macro referenced in scrap 18.

"**numericLinearAim.m**" 18 \equiv

```
<numericLinearAim 17>
```

◇

3 Test Suite

"numericLinearAimTest.m" 19 ≡

```
If [numericAR[{{0,0,0}}]==$rankDeficiency[{{}, {0, 0, 0}}]],  
"passed zero row detection#00 test",  
"failed zero row detection#00 test"]
```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

"numericLinearAimTest.m" 20 ≡

```
If [numericAR[{{a1,b1,c1,d1,e1,f1},{a1,b1,c1,d1,e1,f1}}]  
===  
$rankDeficiency[{{}, {{a1/e1, b1/e1, c1/e1, d1/e1, 1, f1/e1},  
{0, 0, 0, 0, 0, 0}}]],  
"passed zero row detection#01 test",  
"failed zero row detection#01 test"]
```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

"numericLinearAimTest.m" 21 ≡

```
If [numericAR[{{a1,b1,0,0,0,0},{0,0,0,0,a1,b1}}]  
===  
$rankDeficiency[{{a1, b1, 0, 0}, {0, 0, a1, b1}},  
{0, 0, 0, 0, 1, b1/a1}, {0, 0, 0, 0, 0, 0}}]],  
"passed zero row detection#02 test",  
"failed zero row detection#02 test"]
```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

"numericLinearAimTest.m" 22 ≡

```
If [numericAR[{{a,b,c}}]=={{}, {{a/c, b/c, 1}}}},  
"passed AR computation#00 test",  
"failed AR computation#00 test"]
```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

```

"numericLinearAimTest.m" 23 ≡

If [numericAR[{{a1,b1,c1,d1,e1,f1},{a2,b2,c2,d2,e2,f2}}]
===
{{}, {{(a2*f1)/(e2*f1 - e1*f2) + (a1*f2)/(-(e2*f1) + e1*f2),
(b2*f1)/(e2*f1 - e1*f2) + (b1*f2)/(-(e2*f1) + e1*f2),
(c2*f1)/(e2*f1 - e1*f2) + (c1*f2)/(-(e2*f1) + e1*f2),
(d2*f1)/(e2*f1 - e1*f2) + (d1*f2)/(-(e2*f1) + e1*f2),
(e2*f1)/(e2*f1 - e1*f2) + (e1*f2)/(-(e2*f1) + e1*f2),
(f1*f2)/(e2*f1 - e1*f2) + (f1*f2)/(-(e2*f1) + e1*f2)},
{(a2*e1)/(-(e2*f1) + e1*f2) - (a1*e2)/(-(e2*f1) + e1*f2),
(b2*e1)/(-(e2*f1) + e1*f2) - (b1*e2)/(-(e2*f1) + e1*f2),
(c2*e1)/(-(e2*f1) + e1*f2) - (c1*e2)/(-(e2*f1) + e1*f2),
(d2*e1)/(-(e2*f1) + e1*f2) - (d1*e2)/(-(e2*f1) + e1*f2), 0,
-((e2*f1)/(-(e2*f1) + e1*f2)) + (e1*f2)/(-(e2*f1) + e1*f2)}},
"passed AR computation#01 test",
"failed AR computation#01 test"]

```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

```
"numericLinearAimTest.m" 24 ≡
```

```

If [numericAR[{{a1,b1,c1,d1,e1,f1},{a2,b2,c2,d2,0,0}}]
===
{{{a2, b2, c2, d2}}, {{-((a1*d2)/(-(d2*e1) + c2*f1)),
-((b1*d2)/(-(d2*e1) + c2*f1)),
-((c1*d2)/(-(d2*e1) + c2*f1)) + (a2*f1)/(-(d2*e1) + c2*f1),
-((d1*d2)/(-(d2*e1) + c2*f1)) + (b2*f1)/(-(d2*e1) + c2*f1),
-((d2*e1)/(-(d2*e1) + c2*f1)) + (c2*f1)/(-(d2*e1) + c2*f1), 0},
{(a1*c2)/(-(d2*e1) + c2*f1), (b1*c2)/(-(d2*e1) + c2*f1),
(a2*e1)/(d2*e1 - c2*f1) + (c1*c2)/(-(d2*e1) + c2*f1),
(b2*e1)/(d2*e1 - c2*f1) + (c2*d1)/(-(d2*e1) + c2*f1),
(c2*e1)/(d2*e1 - c2*f1) + (c2*e1)/(-(d2*e1) + c2*f1),
(d2*e1)/(d2*e1 - c2*f1) + (c2*f1)/(-(d2*e1) + c2*f1)}},
"passed AR computation#02 test",
"failed AR computation#02 test"]

```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

```

"numericLinearAimTest.m" 25 ≡

If[numericAR[{{a1,b1,c1,d1,e1,f1},{a2,b2,c2,d2,2*e1,2*f1}}]
===
{{{a1 - a2/2, b1 - b2/2, c1 - c2/2, d1 - d2/2}},
{{{(a2*(-2*d1*e1 + d2*e1))/(2*e1*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)),
(b2*(-2*d1*e1 + d2*e1))/(2*e1*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)),
(c2*(-2*d1*e1 + d2*e1))/(2*e1*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)) +
(2*(a1 - a2/2)*f1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1),
(d2*(-2*d1*e1 + d2*e1))/(2*e1*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)) +
(2*(b1 - b2/2)*f1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1) +
(-2*d1*e1 + d2*e1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1) +
(2*(c1 - c2/2)*f1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1),
(2*(d1 - d2/2)*f1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1) +
((-2*d1*e1 + d2*e1)*f1)/(e1*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1))},
{{(a2*(2*c1 - c2))/(2*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)),
(b2*(2*c1 - c2))/(2*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)),
((2*c1 - c2)*c2)/(2*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)) +
(2*(a1 - a2/2)*e1)/(2*d1*e1 - d2*e1 - 2*c1*f1 + c2*f1),
(2*c1 - c2)*d2)/(2*(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1)) +
(2*(b1 - b2/2)*e1)/(2*d1*e1 - d2*e1 - 2*c1*f1 + c2*f1),
((2*c1 - c2)*e1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1) +
(2*(c1 - c2/2)*e1)/(2*d1*e1 - d2*e1 - 2*c1*f1 + c2*f1),
((2*c1 - c2)*f1)/(-2*d1*e1 + d2*e1 + 2*c1*f1 - c2*f1) +
(2*(d1 - d2/2)*e1)/(2*d1*e1 - d2*e1 - 2*c1*f1 + c2*f1)}},
"passed AR computation#03 test",
"failed AR computation#03 test"]

```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

```

"numericLinearAimTest.m" 26 ≡

If[
  numericBiDirectionalAR[{{a1,b1,c1,d1,0,0},{0,0,c2,d2,e2,f2}}]==
  {{{{a1, b1, c1, d1}}, {{0, 0,
    -((c2*d1)/(-(d1*e2) + c1*f2)) + (a1*f2)/(-(d1*e2) + c1*f2),
    -((d1*d2)/(-(d1*e2) + c1*f2)) + (b1*f2)/(-(d1*e2) + c1*f2),
    -((d1*e2)/(-(d1*e2) + c1*f2)) + (c1*f2)/(-(d1*e2) + c1*f2), 0},
    {0, 0, (a1*e2)/(d1*e2 - c1*f2) + (c1*c2)/(-(d1*e2) + c1*f2),
      (b1*e2)/(d1*e2 - c1*f2) + (c1*d2)/(-(d1*e2) + c1*f2),
      (c1*e2)/(d1*e2 - c1*f2) + (c1*e2)/(-(d1*e2) + c1*f2),
      (d1*e2)/(d1*e2 - c1*f2) + (c1*f2)/(-(d1*e2) + c1*f2)}},
    {{{c2, d2, e2, f2}}, {{0, -(b1*c2)/(-(b1*c2) + a1*d2)) +
      (a1*d2)/(-(b1*c2) + a1*d2),
      -((c1*c2)/(-(b1*c2) + a1*d2)) + (a1*e2)/(-(b1*c2) + a1*d2),
      -((c2*d1)/(-(b1*c2) + a1*d2)) + (a1*f2)/(-(b1*c2) + a1*d2), 0, 0},
    {(b1*c2)/(b1*c2 - a1*d2) + (a1*d2)/(-(b1*c2) + a1*d2),
      (b1*d2)/(b1*c2 - a1*d2) + (b1*d2)/(-(b1*c2) + a1*d2),
      (c1*d2)/(-(b1*c2) + a1*d2) + (b1*e2)/(b1*c2 - a1*d2),
      (d1*d2)/(-(b1*c2) + a1*d2) + (b1*f2)/(b1*c2 - a1*d2), 0, 0}}},
    "passed AR computation#04 test",
    "failed AR computation#04 test"]

```

◇

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

```

"numericLinearAimTest.m" 27 ≡

Module[{x,y,z},
With[{symBi=numericBiDirectionalAR[{{a1,b1,c1,d1,0,0},{0,0,c2,d2,e2,f2}}]},,
With[{res=numericSqueeze[Join[symBi[[1,1]],symBi[[2,1]],symBi[[1,2]]],
shouldBe=
{{{0, 0, 1, 0}, {0, 0, 0, 1},
{0, 0, (c2*d1)/(-(d1*e2) + c1*f2) - (a1*f2)/(-(d1*e2) + c1*f2),
(d1*d2)/(-(d1*e2) + c1*f2) - (b1*f2)/(-(d1*e2) + c1*f2)},
{0, 0, -(a1*e2)/(d1*e2 - c1*f2) - (c1*c2)/(-(d1*e2) + c1*f2),
-(b1*e2)/(d1*e2 - c1*f2) - (c1*d2)/(-(d1*e2) + c1*f2)},
Function[{lamVal$10, evcs9, evcs10}, {{0, 0, evcs9, evcs10}}],
{{{c2*d1)/(-(d1*e2) + c1*f2) - (a1*f2)/(-(d1*e2) + c1*f2),
(d1*d2)/(-(d1*e2) + c1*f2) - (b1*f2)/(-(d1*e2) + c1*f2)},
{-(a1*e2)/(d1*e2 - c1*f2) - (c1*c2)/(-(d1*e2) + c1*f2),
-(b1*e2)/(d1*e2 - c1*f2) - (c1*d2)/(-(d1*e2) + c1*f2)}},
If[And[res[[2]][a,b,c]==shouldBe[[2]][a,b,c],Max[Abs[Flatten[res[[1,3]]]-shouldBe[[1,3]]]]]==
"passed squeeze computation#01 test",
"failed squeeze computation#01 test"]]]]

trialSubsN={alpha->2,gamma->1/10,theta->-1/5};
trialSubsNN=N[{alpha->2,gamma->1/10,theta->-1/5}];

simsh= {{0, 0, 0, 0, 0, 0, -1, alpha, 1, -1/2, 0, 0, 0, 0, -1/2},
{0, 0, -1/2, 0, 0, 0, 0, -1/2, 1, 0, 0, 0, 0, 0, 0},
{0, 0, -theta, 0, 0, -1, 0, 1, 0, -gamma, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}}
{{{af,hf},{ab,hb}}}=numericBiDirectionalAR[simsh/.trialSubsNN]

transMat=numericTransitionMatrix[hf];
{tog,j0,pi,lilTransMat}=numericSqueeze[Join[af,ab],transMat];
Eigenvalues[transMat]
Eigenvalues[lilTransMat]

bles=Eigensystem[Transpose[transMat]];
ules=Eigensystem[Transpose[lilTransMat]];
mapper=numericMapper[pi,j0];

ubigEv=numericParticularLam[ules,1,mapper,tog];
ubigEvs=numericEvExtend[DiagonalMatrix[ules[[1]]],ules[[2]],mapper,tog];

```

◊

File defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.

4 Files

```
"numericLinearAim.m" Defined by scrap 18.  
"numericLinearAimTest.m" Defined by scraps 19, 20, 21, 22, 23, 24, 25, 26, 27.
```

5 Macros

```
<kronMathematica 16> Referenced in scrap 17.  
<numericAIMVersionMathematica 7> Referenced in scrap 17.  
<numericARMathematica 5> Referenced in scrap 17.  
<numericAnnihilateRowsMathematica 4> Referenced in scrap 17.  
<numericAsymptoticConstraintMathematica 8> Referenced in scrap 17.  
<numericBiDirectionalARMathematica 6> Referenced in scrap 17.  
<numericComputeAnnihilatorMathematica 3> Referenced in scrap 17.  
<numericEliminateInessentialLagsMathematica 10> Referenced in scrap 17.  
<numericEvExtendMathematica 13> Referenced in scrap 17.  
<numericExtendToBasisMathematica 15> Referenced in scrap 17.  
<numericLinearAim 17> Referenced in scrap 18.  
<numericMapperMathematica 12> Referenced in scrap 17.  
<numericParticularLamMathematica 14> Referenced in scrap 17.  
<numericRightMostAllZeroQMathematica 1> Referenced in scrap 17.  
<numericShiftRightAndRecordMathematica 2> Referenced in scrap 17.  
<numericSqueezeMathematica 11> Referenced in scrap 17.  
<numericTransitionMatrix 9> Referenced in scrap 17.
```

6 Identifiers

```
$zeroTol: 1, 3.  
kronMathematica: 16, 17.  
numericAnnihilateRowsMathematica: 4, 17.  
numericARMathematica: 5, 17.  
numericBiDirectionalARMathematica: 6, 17.  
numericComputeAnnihilatorMathematica: 3, 17.  
numericEvExtendMathematica: 13, 17.  
numericExtendToBasisMathematica: 15, 17.  
numericLinearAim: 17, 18.  
numericMapperMathematica: 12, 17.  
numericParticularLamMathematica: 14, 17.  
numericRightMostAllZeroQMathematica: 1, 17.  
numericShiftRightAndRecordMathematica: 2, 17.  
numericSqueeze: 8, 11, 27.  
numericTransitionMatrix: 9, 17, 27.
```

? ? . ? ?? ?

References

- [1] Gary Anderson. Continuous time application of the anderson-moore(aim) algorithm for imposing the saddle point property in dynamic models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://irmum1.frb.gov/~m1gsa00/summariesAbstracts.html>, 1997.
- [2] Gary Anderson. A parallel programming implementation of the anderson-moore(aim) algorithm. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://irmum1.frb.gov/~m1gsa00/summariesAbstracts.html>, 1997.
- [3] Gary Anderson and George Moore. An efficient procedure for solving linear perfect foresight models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://irmum1.frb.gov/~m1gsa00/summariesAbstracts.html>, 1983.
- [4] Gary Anderson and George Moore. A linear algebraic procedure for solving linear perfect foresight models. *Economics Letters*, 17, 1985.
- [5] Michael W. Berry. Large scale sparse singular value computations. University of Tennessee, Department of Computer Science, 1996.
- [6] Olivier Jean Blanchard and C. Kahn. The solution of linear difference models under rational expectations. *Econometrica*, 48, 1980.
- [7] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins, 1989.
- [8] E. V. Krishnamurthy. *Parallel Processing: Principles and Practice*. Addison-Wesley, 1989.
- [9] Ben Noble. *Applied Linear Algebra*. Prentice-Hall, Inc., 1969.
- [10] J. Taylor. Conditions for unique solutions in stochastic macroeconomic models with rational expectations. *Econometrica*, 45:1377–1385, —SEP— 77.
- [11] C. H. Whiteman. *Linear Rational Expectations Models: A User’s Guide*. University of Minnesota, 1983.